

Approximate Multi-Matroid Intersection via Iterative Refinement^{*}

André Linhares^{†1}, Neil Olver^{‡2}, Chaitanya Swamy^{†1}, and Rico Zenklusen^{§3}

¹ Dept. of Combinatorics and Optimization, Univ. Waterloo, Waterloo, Canada.
{alinhare, cswamy}@uwaterloo.ca

² Dept. of Econometrics and Operations Research, Vrije Universiteit Amsterdam,
Amsterdam, The Netherlands; and CWI, Amsterdam, The Netherlands.
n.olver@vu.nl

³ Department of Mathematics, ETH Zurich, Zurich, Switzerland. ricoz@math.ethz.ch

Abstract. We introduce a new iterative rounding technique to round a point in a matroid polytope subject to further matroid constraints. This technique returns an independent set in one matroid with limited violations of the constraints of the other matroids. In addition to the classical steps of iterative relaxation approaches, we iteratively *refine* involved matroid constraints. This leads to more restrictive constraint systems whose structure can be exploited to prove the existence of constraints that can be dropped. Hence, throughout the iterations, we both tighten constraints and later relax them by dropping constraints under certain conditions. Due to the refinement step, we can deal with considerably more general constraint classes than existing iterative relaxation and rounding methods, which typically involve a single matroid polytope with additional simple cardinality constraints that do not overlap too much. We show that our rounding method, combined with an application of a matroid intersection algorithm, yields the first 2-approximation for finding a maximum-weight common independent set in 3 matroids.⁴ Moreover, our 2-approximation is LP-based and settles the integrality gap for the natural relaxation of the problem. Prior to our work, no upper bound better than 3 was known for the integrality gap, which followed from the greedy algorithm. We also discuss various other applications of our techniques, including an extension that allows us to handle a mixture of matroid and knapsack constraints.

1 Introduction

Matroids are among the most fundamental and well-studied structures in combinatorial optimization. Recall that a *matroid* is a pair $M = (N, \mathcal{I})$, where N

^{*} A preliminary version [21] appeared in the Proceedings of the 20th IPCO 2019.

[†] Research supported by NSERC grant 327620-09 and an NSERC DAS Award.

[‡] Supported by NWO VIDI grant 016.Vidi.189.087.

[§] Supported by Swiss National Science Foundation grant 200021_165866.

⁴ We follow the convention that an α -approximation for a maximization problem returns a solution of value at least a $1/\alpha$ fraction of the optimum.

is a finite ground set and $\mathcal{I} \subseteq 2^N$ is a family of sets, called *independent sets*, such that (i) $\emptyset \in \mathcal{I}$, (ii) if $A \in \mathcal{I}$ and $B \subseteq A$, then $B \in \mathcal{I}$, and (iii) if $A, B \in \mathcal{I}$ with $|A| > |B|$, then there is an element $e \in A \setminus B$ such that $B \cup \{e\} \in \mathcal{I}$. We make the standard assumption that a matroid is specified via an *independence oracle*, which, given $S \subseteq N$ as input, returns if $S \in \mathcal{I}$. Matroids capture many interesting problems, and matroid-optimization algorithms provide a powerful tool in the design and analysis of efficient algorithms. A key matroid optimization problem is *matroid intersection*, wherein we seek a maximum-weight set that is independent in *two* matroids. Various efficient algorithms are known for matroid intersection, and we also have a celebrated min-max theorem and a polyhedral understanding of the problem. The versatility of matroid intersection comes from the fact that the intersection of matroids allows for describing a very broad family of constraints.

Unfortunately, as soon as the intersection of 3 or more matroids is considered, already the unweighted version of determining a maximum cardinality common independent set becomes *APX-hard*. Due to its fundamental nature, and many natural special cases, the problem of optimizing over 3 or more matroids has received considerable attention. In particular, there is extensive prior work on maximum cardinality problems [18, Chapter 5], maximization of submodular functions over the intersection of multiple matroids (see [7, 11, 14, 19, 20] and the references therein), and various interesting special cases like k -dimensional matching (see [5, 9, 10, 15, 16] and the references therein; many of these results also apply to the k -set packing problem, which generalizes k -dimensional matching).

Nevertheless, there are still basic open questions regarding the approximability of the optimization over 3 or more matroids. Perhaps the most basic problem of this type is the *weighted 3-matroid intersection* problem, defined as follows.

Weighted 3-matroid intersection. Given matroids $M_i = (N, \mathcal{I}_i)$, for $i = 1, 2, 3$, on a common ground set N , and a weight vector $w \in \mathbb{R}^N$, solve

$$\max \{w(I) : I \in \mathcal{I}_1 \cap \mathcal{I}_2 \cap \mathcal{I}_3\},$$

where we use the shorthand $w(S) := \sum_{e \in S} w(e)$ for any set $S \subseteq N$.

The *unweighted 3-matroid intersection* problem, which is also sometimes called the *cardinality version* of 3-matroid intersection, is the special case where $w(e) = 1$ for all $e \in N$, so $w(S) = |S|$ for $S \subseteq N$.

The 3-matroid intersection problem has the following natural and canonical linear programming relaxation:

$$\max \{w^T x : x \in P_{\mathcal{I}_1} \cap P_{\mathcal{I}_2} \cap P_{\mathcal{I}_3}\}. \quad (\text{LP}_{3\text{-mat}})$$

Here, $P_{\mathcal{I}} \subseteq [0, 1]^N$ denotes the matroid polytope of a matroid $M = (N, \mathcal{I})$, i.e., the convex hull of all characteristic vectors of sets in \mathcal{I} . It has a well-known inequality description given by

$$P_{\mathcal{I}} = \{x \in \mathbb{R}_{\geq 0}^N : x(S) \leq r_M(S) \ \forall S \subseteq N, S \neq \emptyset\},^5$$

⁵ We exclude the trivially satisfied constraint corresponding to $S = \emptyset$ to highlight that, when referring later to a constraint of $P_{\mathcal{I}}$ that is tight for some point $x \in \mathbb{R}^N$, then such a constraint never corresponds to $S = \emptyset$.

where $r_M : 2^N \rightarrow \mathbb{Z}_{\geq 0}$ is the *rank function* of M , defined by $r_M(S) := \max\{|I| : I \in \mathcal{I}, I \subseteq S\}$ for any $S \subseteq N$. The rank function is submodular, and $r_M(S)$ can be computed for any $S \subseteq N$ using an independence oracle. It will therefore often be convenient to assume that a matroid M is specified via its *rank oracle* that, given $S \subseteq N$ as input, returns $r_M(S)$. In particular, one can efficiently optimize any linear function over $P_{\mathcal{I}}$ given a rank oracle (or equivalently an independence oracle). The above LP-relaxation extends naturally to the *k-matroid intersection* problem, which is the extension of 3-matroid intersection to k matroids.

Whereas $(\text{LP}_{3\text{-mat}})$, and its extension $(\text{LP}_{k\text{-mat}})$ to k -matroid intersection, are well-known LP-relaxations, there remain various gaps in our understanding of these relaxations. It is widely known that the greedy algorithm is a k -approximation for k -matroid intersection. Moreover, this approximation is relative to $\text{OPT}_{\text{LP}_{k\text{-mat}}}$ (the optimal value of $(\text{LP}_{k\text{-mat}})$), which leads to the current best upper bound of k on the integrality gap of $(\text{LP}_{k\text{-mat}})$, for all $k \geq 3$. Interestingly, there is no example of matroids known for which $(\text{LP}_{k\text{-mat}})$ has integrality gap strictly larger than $k - 1$. A lower bound on the integrality gap of $k - 1$ is achievable when all involved matroids are unitary partition matroids⁶ and $k - 1$ is a prime power [12].

Significant progress on approximating k -matroid intersection was achieved by Lee, Sviridenko, and Vondrák [20], who presented, for any fixed $\epsilon > 0$, a $(k - 1 + \epsilon)$ -approximation, based on local search. Unfortunately, apart from the fact that their method has a running time depending exponentially in ϵ^{-1} , it does not shed any insights on $(\text{LP}_{k\text{-mat}})$, because the above guarantee is not relative to the optimum value $\text{OPT}_{\text{LP}_{k\text{-mat}}}$ of $(\text{LP}_{k\text{-mat}})$. Further progress on understanding the quality of the LP-relaxations has only been achieved in special cases. In particular, for unweighted 3-matroid intersection, a result by Aharoni and Berger [1] implies that the integrality gap of $(\text{LP}_{3\text{-mat}})$ with w being the all-ones vector is at most 2. Unfortunately, the purely combinatorial proof technique of Aharoni and Berger does not seem to have any direct implications to the weighted case, nor is the result algorithmic. More recently, for *unweighted k-matroid intersection*, Lau, Ravi, and Singh [18, Chapter 5] presented an algorithmic counterpart for the unweighted case for the intersection of any number k of matroids. More precisely, they give an LP-based $(k - 1)$ -approximation through iterative rounding. Their proof is based on iteratively identifying an element with “large” fractional value, picking it, and altering the fractional solution so that it remains feasible; the last step crucially uses the fact that the instance is unweighted to control the loss in the LP objective value. For the intersection of k unitary partition matroids, a problem also known as *k-dimensional matching*, Chan and Lau [5] obtained a $(k - 1)$ -approximation based on $(\text{LP}_{k\text{-mat}})$, and Parekh and Pritchard [22] later obtained the same approximation factor for the intersection of k (not necessarily unitary) partition matroids. Notice that whenever $k - 1$ is a prime power, these results show, together with the above-mentioned integrality gap lower bound,

⁶ In a partition matroid, the constraints are bounds on the number of items selected from each part in a given partition. If these bounds are all unit, the partition matroid is called unitary.

that $k - 1$ is indeed the integrality gap of $(\text{LP}_{k\text{-mat}})$ if all involved matroids are partition matroids.

Although it is generally believed that a $(k - 1)$ -approximation for k -matroid intersection should exist, and that the integrality gap of $(\text{LP}_{k\text{-mat}})$ is equal to the known lower bound of $k - 1$, this has remained open even for 3-matroid intersection (prior to our work). Recall that in this case, the best known upper and lower bounds on the integrality gap of $(\text{LP}_{3\text{-mat}})$ are 3 (via the classical greedy algorithm) and 2, respectively. Moreover, the only method known to beat the trivial 3-approximation of the greedy algorithm is the non-LP based and computationally quite expensive $(2 + \epsilon)$ -approximation in [20]. One main reason for the limited progress is the lack of techniques for rounding points in the intersection of multiple matroid polytopes with sufficiently strong properties. In particular, one technical difficulty that is encountered is that the tight constraints (even at an extreme point) may have large overlap.

1.1 Our results.

We introduce a new iterative rounding approach to handle the above difficulties that allows for dealing with a very general class of optimization problems involving matroids. Before delving into the details of this technique, we highlight its main implication in the context of 3-matroid intersection.

Theorem 1. *There is an efficient algorithm that, given any instance of the 3-matroid intersection problem, returns a common independent set R with $w(R) \geq \frac{1}{2} \text{OPT}_{\text{LP}_{3\text{-mat}}}$.*

This is the first 2-approximation for 3-matroid intersection (with general weights). Moreover, our result settles the integrality gap of $(\text{LP}_{3\text{-mat}})$, since it matches the known integrality gap lower bound of 2.

The chief new technical ingredient that leads to Theorem 1, and results for other applications discussed in Section 3, is an approximation result based on a novel *iterative refinement* technique (see Section 2.1). We will apply this technique to the following more general problem, from which our results for 3-matroid intersection will be easily derived.

Let $N = N_0$ be a finite ground set, $M_i = (N_i, \mathcal{I}_i)$ for $i = 0, \dots, k$ be $k + 1$ matroids, where $N_i \subseteq N$, and $w \in \mathbb{R}^N$ be a weight vector (note that *negative* weights are allowed). We consider the problem

$$\max \{w(I) : I \in \mathcal{B}_0, \quad I \cap N_i \in \mathcal{I}_i \quad \forall i \in [k]\}, \quad (1)$$

where \mathcal{B}_0 is the set of all bases of M_0 and $[k] := \{1, \dots, k\}$. We consider matroids M_i for $i \in [k]$ defined on ground sets N_i that are subsets of N because, as we show below, we obtain guarantees depending on how strongly the sets N_i overlap; intuitively, problem (1) becomes easier as the overlap between N_1, \dots, N_k decreases, and our guarantee improves correspondingly.

We cannot hope to solve (1) optimally, as this would enable one to solve the NP-hard k -matroid intersection problem. Our goal will be to find a basis of M_0 of large weight that is “approximately independent” in the matroids M_1, \dots, M_k .

How should “approximate independence” be quantified? Perhaps the two notions that first come to mind are additive and multiplicative violation of the rank constraints. Whereas additive violations are common in the study of degree-bounded MST problems, which can be cast as special cases of (1), it turns out that such a guarantee is impossible to obtain (in polytime) for (1). More precisely, we show in Section 4 (via a replication idea) that, even for $k = 2$, if we could find in polytime a basis B of M_0 satisfying $|B| \leq r_{M_i}(B) + \alpha$ for $i = 1, 2$, where $\alpha = O(|N|^{1-\epsilon})$ for any $\epsilon > 0$, then we could efficiently find a basis of M_0 that is independent in M_1, M_2 ; the latter problem is easily seen to be NP -hard via a reduction from Hamiltonian path. We therefore consider multiplicative violation of the rank constraints. Given some $\alpha \geq 1$, we say that $S \subseteq N$ is α -independent for a matroid $M = (N, \mathcal{I})$, if $|T| \leq \alpha \cdot r_M(T) \forall T \subseteq S$ (equivalently, $\chi^S \in \alpha P_{\mathcal{I}}$, where χ^S is the characteristic vector of S). This is much stronger than simply requiring that $|S| \leq \alpha \cdot r_M(S)$, and it is easy to give examples where this weaker notion admits sets that one would consider to be quite far from being independent. An appealing feature of the stronger definition is that, using the min-max result for matroid intersection (or via matroid partition; see, e.g., [8, Chapter 8]), it follows easily that if $\alpha \in \mathbb{Z}_{\geq 1}$, then S is α -independent if and only if S can be partitioned into α independent sets of M . We now state the guarantee we obtain for (1) precisely. We consider the following canonical LP relaxation of (1):

$$\max \{w^T x : x \in \mathbb{R}_{\geq 0}^N, \quad x \in P_{\mathcal{B}_0}, \quad x|_{N_i} \in P_{\mathcal{I}_i} \quad \forall i \in [k]\}, \quad (\text{LP}_{\text{mat}})$$

where for a set $S \subseteq N$, we use $x|_S \in \mathbb{R}^S$ to denote the restriction of x to S , and $P_{\mathcal{B}_0} := P_{\mathcal{I}_0} \cap \{x \in \mathbb{R}^N : x(N) = r_{M_0}(N)\}$ is the matroid base polytope of M_0 . For ease of notation, we will sometimes omit an explicit restriction to the relevant ground set when this can cause no ambiguity; thus we may write $x \in P_{\mathcal{I}_i}$ instead of $x|_{N_i} \in P_{\mathcal{I}_i}$; $R \in \mathcal{I}_i$ instead of $R \cap N_i \in \mathcal{I}_i$; and “ R is α -independent in M_i ” instead of “ $R \cap N_i$ is α -independent in M_i ”.

Our main result for (1), based on a new iterative rounding algorithm for (LP_{mat}) described in Section 2, is the following.

Theorem 2. *Let $q_1, \dots, q_k \in \mathbb{Z}_{\geq 1}$ such that*

$$\sum_{i \in [k]: e \in N_i} q_i^{-1} \leq 1 \quad \forall e \in N. \quad (2)$$

If (LP_{mat}) is feasible, then one can efficiently compute $R \subseteq N$ such that

- (i) $w(R) \geq \text{OPT}_{\text{LP}_{\text{mat}}}$;
- (ii) $R \in \mathcal{B}_0$; and
- (iii) R is q_i -independent in M_i for all $i \in [k]$.

Note that, in particular, taking $q_i = \max_{e \in N} |\{j \in [k] : e \in N_j\}|$ for all $i \in [k]$ satisfies (2). Thus, we violate the constraints imposed by the other matroids M_1, \dots, M_k by a multiplicative factor depending on how strongly the N_i s overlap.

While we have stated Theorem 2 in terms of *bases* of M_0 , the following natural variant is easily deduced from it and, as we will show subsequently, readily implies our main result, Theorem 1.

Corollary 3. *Theorem 2 also holds when R is required only to be an independent set in M_0 (as opposed to a basis), and we replace $P_{\mathcal{B}_0}$ in (LP_{mat}) by $P_{\mathcal{I}_0}$.*

Proof. We modify M_0 to obtain a matroid \widehat{M}_0 on the ground set $N_0 \cup F$, where F is a set of $r_M(N_0)$ additional elements with 0 weight. We define the rank function of \widehat{M}_0 as $\widehat{r}(S) := \min\{r_M(S \cap N_0) + |S \cap F|, r_M(N_0)\}$. That is, \widehat{M}_0 is the union of M_0 with a free matroid on F , but then truncated to have rank $r_{M_0}(N_0)$. Let $P_{\widehat{\mathcal{B}}_0}$ be the matroid base polytope of \widehat{M}_0 . It is now easy to see that if $x \in \mathbb{R}^{N_0 \cup F}$ lies in $P_{\widehat{\mathcal{B}}_0}$, then $x|_{N_0} \in P_{\mathcal{I}_0}$. Moreover, we can extend $x \in \mathbb{R}^{N_0}$ with $x \in P_{\mathcal{I}_0}$ to $x' \in \mathbb{R}^{N_0 \cup F}$ so that $x' \in P_{\widehat{\mathcal{B}}_0}$ and $x'|_{N_0} = x$. The corollary thus follows by applying Theorem 2 to $\widehat{M}_0, M_1, \dots, M_k$. \square

A variety of problem settings can be handled via Theorem 2 and Corollary 3 in a unified way. We first show how to obtain a crisp, simple proof of Theorem 1.

Proof of Theorem 1. Given matroids $M_i = (N, \mathcal{I}_i)$ for $i = 0, 1, 2$, and a weight vector $w \in \mathbb{R}^N$, we first solve $(\text{LP}_{3\text{-mat}})$ to obtain an optimal solution x^* . Now we utilize Corollary 3 with the same three matroids, and $q_1 = q_2 = 2$. Clearly, these q -values satisfy (2), and x^* is a feasible solution to (LP_{mat}) , when we replace $P_{\mathcal{B}_0}$ by $P_{\mathcal{I}_0}$. Thus, we obtain a set $A \in \mathcal{I}_0$ with $w(A) \geq w^T x^*$ and $\chi^A \in 2P_{\mathcal{I}_1} \cap 2P_{\mathcal{I}_2}$.

It is well known that $P_{\mathcal{I}_1} \cap P_{\mathcal{I}_2}$ is a polytope with integral extreme points (see, e.g., [8, Chapter 8]), known as the *matroid intersection polytope*. Since $\chi^A/2 \in P_{\mathcal{I}_1} \cap P_{\mathcal{I}_2}$, by using an algorithm for (weighted) matroid intersection applied to matroids M_1 and M_2 restricted to A we can find a set $R \subseteq A$ such that $R \in \mathcal{I}_1 \cap \mathcal{I}_2$ and $w(R) \geq w^T \chi^A/2 \geq w^T x^*/2$. Finally, since $R \subseteq A$ and $A \in \mathcal{I}_0$, we also have that $R \in \mathcal{I}_0$. \square

Beyond 3-matroid intersection, Theorem 2 is applicable to various constrained (e.g., degree-bounded) spanning tree problems; we expand on this below. In Section 3, we discuss an application in this direction, wherein we seek a min-cost spanning tree satisfying matroid independence constraints on cuts defined by a given collection of pairwise disjoint node-sets. Using Theorem 2, we obtain a spanning tree with a multiplicative factor-2 violation of the matroid constraints.

In Section 3.2, we also present a noteworthy extension of Theorem 2 that allows one to handle both matroid independence and *knapsack constraints*. Suppose in addition to the constraints imposed by the $k+1$ matroids M_0, M_1, \dots, M_k , we also impose t additional *knapsack constraints*. We obtain a basis R of M_0 that satisfies the other (matroid independence and knapsack) constraints approximately, where we can trade off the violation of all involved constraints in a manner similar to Theorem 2 (see Theorem 10 and its corollaries). By way of comparison, Chekuri et al. [7] obtain (among other results) an $O(k+t)$ -approximation for this setting with no constraint violation.

1.2 Related work and connections.

By choosing M_0 to be a graphic matroid, problem (1) generalizes many known constrained spanning tree problems, including degree-bounded spanning trees,

and generalizations thereof considered by Bansal et al. [3], Király et al. [17], and Zenklusén [25]. Theorem 2 thus yields a unified way to deal with various spanning tree problems considered in the literature, where the degree constraints are violated by at most a constant factor. However, as noted earlier, whereas the above works obtain stronger, additive violation results, such guarantees are not possible for our general problem (1) as we show in Section 4. This hardness of obtaining small additive violations carries over to the spanning tree application that we consider in Section 3 (which generalizes the matroidal degree-bounded spanning tree problem considered in [25]).

To showcase how Theorem 2 can be used for such problems, consider the minimum degree-bounded spanning tree problem, where given is a graph $G = (V, E)$ with edge weights $w : E \rightarrow \mathbb{R}$ and degree bounds $B_v \in \mathbb{Z}_{\geq 1}$ for $v \in V$. The nominal problem asks to find a spanning tree $T \subseteq E$ with $|T \cap \delta(v)| \leq B_v$ for $v \in V$ minimizing $w(T)$, where $\delta(v)$ denotes the set of edges incident with v . Here one can apply Theorem 2 with M_0 being the graphic matroid of G , and for each $v \in V$ we define a uniform matroid M_v with ground set $\delta(v)$ and rank B_v . Theorem 2 with $q_v = 2 \forall v \in V$ and negated edge weights leads to a spanning tree T with $|T \cap \delta(v)| \leq 2B_v \forall v \in V$ and weight no more than the optimal LP-weight. Whereas this is a simple showcase example, Theorem 2 can be used in a similar way for considerably more general constraints than just degree constraints.

Finally, we highlight a main difference of our approach compared to prior techniques. Prior techniques for related problems, as used for example by Singh and Lau [24], Király et al. [17], and Bansal et al. [3], successively drop constraints of a relaxation. Also, interesting variations have been suggested that do not just drop constraints but may relax constraints by replacing a constraint by a weaker family (see work by Bansal et al. [2]). In contrast, our method does not just relax constraints, but also strengthens the constraint family in some iterations, so as to simplify it and enable one to drop constraints later on.

2 Our rounding technique

Our rounding technique heavily relies on a simple yet very useful “splitting” procedure for matroids, which we call *matroid refinement*.

2.1 Matroid refinement

Let $M = (N, \mathcal{I})$ be a matroid with rank function $r_M : 2^N \rightarrow \mathbb{Z}_{\geq 0}$, and let $S \subsetneq N$, $S \neq \emptyset$. The *refinement* of M with respect to S consists of the matroids $M^{(1)} = M|_S$ obtained by restricting M to S , and $M^{(2)} = M/S$ obtained by contracting S in M . Formally, the independent sets of the two matroids $M^{(1)} = (S, \mathcal{I}^{(1)})$, $M^{(2)} = (N \setminus S, \mathcal{I}^{(2)})$ are given by

$$\begin{aligned} \mathcal{I}^{(1)} &= \{I \subseteq S : I \in \mathcal{I}\} \\ \text{and } \mathcal{I}^{(2)} &= \{I \subseteq N \setminus S : I \cup I_S \in \mathcal{I}\}, \end{aligned}$$

where $I_S \in \mathcal{I}$ is a maximum cardinality independent subset of S . It is well-known that the definition of $\mathcal{I}^{(2)}$ does not depend on which set I_S is chosen. The rank function of $M^{(1)}$ and $M^{(2)}$ are given by

$$\begin{aligned} r_{M^{(1)}}(A) &= r_M(A) & \forall A \subseteq S, \\ \text{and } r_{M^{(2)}}(B) &= r_M(B \cup S) - r_M(S) & \forall B \subseteq N \setminus S. \end{aligned} \quad (3)$$

For a proof of this, and more information on matroid restrictions and contractions, we refer the reader to [23, Volume B, Chapter 39]. The following lemmas describe some basic yet important relations between a matroid $M = (N, \mathcal{I})$ and its refinement. We will use the notation defined above for the remainder of this subsection.

Lemma 4. *If $x \in \mathbb{R}^N$ satisfies $x|_S \in P_{\mathcal{I}^{(1)}}$ and $x|_{N \setminus S} \in P_{\mathcal{I}^{(2)}}$, then $x \in P_{\mathcal{I}}$.*

Proof. For any set $A \subseteq N$, we have

$$\begin{aligned} x(A) &= x(A \cap S) + x(A \setminus S) \\ &\leq r_{M^{(1)}}(A \cap S) + r_{M^{(2)}}(A \setminus S) \\ &= r_M(A \cap S) + r_M(A \cup S) - r_M(S) \\ &\leq r_M(A), \end{aligned}$$

where the last inequality follows from $r_M(A \cap S) + r_M(A \cup S) \leq r_M(A) + r_M(S)$, which holds by submodularity of r_M . Because the above inequality holds for every $A \subseteq N$, we obtain $x \in P_{\mathcal{I}}$ as desired. \square

The description of refinement as replacing a matroid with two disjoint ones will be most convenient for describing and analyzing our algorithm. There is an alternative, more geometric, viewpoint one can take. The direct sum of $M|_S$ and M/S is a matroid $M' = (N, \mathcal{I}')$ with $\mathcal{I}' \subseteq \mathcal{I}$. The matroid base polytope of M' is precisely the face of the matroid base polytope of M defined by the constraint $x(S) = r_M(S)$. The conditions on x in Lemma 4 exactly say that x is in the matroid polytope of M' , and the conclusion of the lemma is that such a point is in the matroid polytope of M . From this perspective, the next lemma shows that any point in the matroid polytope of M which is tight for S is in the matroid polytope of M' . This will help us later to show that, even after one of our (well-chosen) refinement steps in our iterative algorithm, the previous linear programming solution remains feasible.

Lemma 5. *Let $x \in P_{\mathcal{I}}$ be such that $x(S) = r_M(S)$. Then $x|_S \in P_{\mathcal{I}^{(1)}}$ and $x|_{N \setminus S} \in P_{\mathcal{I}^{(2)}}$.*

Proof. Recall that to show that a nonnegative vector $y \in \mathbb{R}_{\geq 0}^{\bar{N}}$ lies inside the matroid polytope of a matroid with rank function \bar{r} , it suffices to check that it satisfies all the rank constraints $y(I) \leq \bar{r}(I)$ for all $I \subseteq \bar{N}$.

For any $I \subseteq S$,

$$x|_S(I) = x(I) \leq r_M(I) = r_{M^{(1)}}(I),$$

and hence it is clear that $x|_S \in P_{\mathcal{I}^{(1)}}$.

To see that $x|_{N \setminus S} \in P_{\mathcal{I}^{(2)}}$, consider any $I \subseteq N \setminus S$. We have

$$\begin{aligned} x(I) &= x(I) + x(S) - r_M(S) \\ &= x(I \cup S) - r_M(S) \\ &\leq r_M(I \cup S) - r_M(S) \\ &= r_{M^{(2)}}(S), \end{aligned}$$

where the first equation is a consequence of $x(S) = r_M(S)$, the inequality is implied by $x \in P_{\mathcal{I}}$, and the last equation holds due to (3). \square

Finally, the next lemma shows that a solution that is nearly independent in both matroids obtained through refinement, is also nearly independent in the original matroid before refinement.

Lemma 6. *Let $\alpha \in \mathbb{Z}_{\geq 1}$. If R_1 is α -independent in $M^{(1)}$ and R_2 is α -independent in $M^{(2)}$, then $R_1 \cup R_2$ is α -independent in M .*

Proof. Let $y = \chi^{R_1 \cup R_2} / \alpha$. Then $y|_S \in P_{\mathcal{I}^{(1)}}$ and $y|_{N \setminus S} \in P_{\mathcal{I}^{(2)}}$, and so $y \in P_{\mathcal{I}}$ by Lemma 4. Thus $\chi^{R_1 \cup R_2} \in \alpha P_{\mathcal{I}}$ as required. \square

Intuitively, matroid refinement serves to partly decouple the matroid independence constraints for M , thereby allowing one to work with somewhat “simpler” matroids subsequently, and we leverage this carefully in our algorithm.

2.2 An algorithm based on iterative refinement and relaxation

Algorithm 1 describes our method to prove Theorem 2. Recall that the input is an instance of problem (1), which consists of $k + 1$ matroids $M_i = (N_i, \mathcal{I}_i)$ for $i = 0, \dots, k$, where each N_i is a subset of a finite ground set $N = N_0$, and a weight vector $w \in \mathbb{R}^N$. We are also given integers $q_i \geq 1$ for $i \in [k]$ satisfying (2).

Algorithm 1 starts by solving the natural LP-relaxation in step 3 to obtain an optimal extreme point x^* . As is common in iterative rounding algorithms, we delete all elements of value 0 and fix all elements of value 1 through contractions in steps 4–5. Apart from these standard operations, we *refine* the matroids in steps 7–9, as long as there is a matroid $M' = (N', \mathcal{I}')$ in our collection \mathcal{M} with a nontrivial x^* -tight set $S \subseteq N'$, i.e., $x^*(S) = r_{M'}(S)$ and $S \notin \{\emptyset, N'\}$. Notice that after the refinement step, the q -values for the matroids in the new collection \mathcal{M} continue to satisfy (2). Step 11 is our relaxation step, where we drop a matroid $M' = (N', \mathcal{I}')$ if $|N'| - r_{M'}(N') \leq q_{M'} - 1$. This is the step that results in a violation of the matroid constraints, but, as we show, the preceding condition ensures that even if we select all elements of N' in the solution, the violation is still within the prescribed bounds. Moreover, we will show in Lemma 7 that, whenever Algorithm 1 is at step 11, there is a matroid satisfying the given conditions that can be dropped. We remark that, in step 11, one could also drop *all* matroids $M' \in \mathcal{M}$ fulfilling this condition, instead of just a single one, without impacting the correctness of the algorithm.

Algorithm 1 Iterative refinement and relaxation algorithm for Theorem 2

-
- (Initialization.)
- 1: $\mathcal{M} \leftarrow \{M_1, \dots, M_k\}$, $\bar{M}_0 = (\bar{N}_0, \bar{\mathcal{I}}_0) \leftarrow M_0$, $q_{M_i} \leftarrow q_i$ for all $i \in [k]$.
 - 2: $R \leftarrow \emptyset$.
- (Start of main loop.)
- 3: Compute an optimal vertex solution x^* to (LP_{mat}) for the matroids $\{\bar{M}_0\} \cup \mathcal{M}$.
- (Deletion and contraction.)
- 4: For each $e \in \bar{N}_0$ with $x^*(e) = 0$, delete e from \bar{M}_0 and all matroids in \mathcal{M} containing e .
 - 5: For each $e \in \bar{N}_0$ with $x^*(e) = 1$, contract e in \bar{M}_0 and all matroids in \mathcal{M} containing e , and add e to R .
 - 6: **if** $\bar{N}_0 = \emptyset$ **then return** R .
- (Refinement.)
- 7: **while** there is a matroid $M' = (N', \mathcal{I}') \in \mathcal{M}$ s.t. $\exists \emptyset \neq S \subsetneq N'$ with $x^*(S) = r_{M'}(S)$ **do**
 - 8: Set $M'_1 = M'|_S$, $M'_2 = M'/S$, and $q_{M'_1} = q_{M'_2} = q_{M'}$.
 - 9: Update $\mathcal{M} \leftarrow (\mathcal{M} \setminus \{M'\}) \cup \{M'_1, M'_2\}$.
 - 10: **end while**
- (Relaxation.)
- 11: Find a matroid $M' = (N', \mathcal{I}') \in \mathcal{M}$, such that $|N'| - r_{M'}(N') \leq q_{M'} - 1$; remove M' from \mathcal{M} .
 - 12: **goto** step 3.
-

In order to find an x^* -tight set $\emptyset \neq S \subsetneq N'$ (if one exists) in step 7, one can, for example, minimize the submodular function $r_{M'}(A) - x^*(A)$ over the sets $\emptyset \neq A \subsetneq N'$. Depending on the matroids involved, faster specialized approaches can be employed.

It is perhaps illuminating to consider the combined effect of all the refinement steps and step 11 corresponding to a given basic optimal solution x^* . Using standard uncrossing techniques (see, e.g., [18, Chapter 5]), one can show that for each matroid $M' = (N', \mathcal{I}') \in \mathcal{M}$, there is a nested family of sets $\emptyset \subsetneq S_1 \subsetneq \dots \subsetneq S_p \subseteq N'$ whose rank constraints span the x^* -tight constraints of M' , and so any S_i can be used to refine M' . The combined effect of all refinements for M' can be seen as replacing M' by the matroids $(M'|_{S_\ell})/S_{\ell-1}$ for $\ell = 1, \dots, p+1$, where $S_0 := \emptyset$, $S_{p+1} := N'$. Step 11 chooses some $M' \in \mathcal{M}$ and a “ring” $S_\ell \setminus S_{\ell-1}$ of its nested family satisfying $|S_\ell \setminus S_{\ell-1}| - x^*(S_\ell \setminus S_{\ell-1}) < q_{M'}$, and drops the matroid created for this ring.

Analysis We first show that the algorithm is well defined, in the sense that a matroid can always be dropped in step 11. Note that refinements guarantee that whenever the algorithm is at step 11, then for any $M' = (N', \mathcal{I}') \in \mathcal{M}$, only the constraint of $P_{\mathcal{I}'}$ corresponding to N' may be x^* -tight. This allows us to leverage ideas similar to those in [3, 17].

Lemma 7. *Each time that step 11 is reached in an execution of Algorithm 1, there exists at least one matroid $M' \in \mathcal{M}$ satisfying the stated conditions and which hence can be dropped.*

Proof. Consider the current collection of matroids \mathcal{M} at the beginning of step 11 in some iteration of the algorithm. (Recall that \mathcal{M} does not contain \bar{M}_0 , the current version of M_0 .) Let x^* be the current basic solution. It satisfies $0 < x^*(e) < 1$ for all $e \in \bar{N}_0$, since elements with $x^*(e) \in \{0, 1\}$ would have been deleted or contracted in steps 4 and 5.

Consider a full-rank subsystem of (LP_{mat}) , $Ax = b$, consisting of linearly independent, x^* -tight constraints. By standard uncrossing arguments (see, e.g., [18, Chapter 5]), we may assume that the constraints of $Ax = b$ coming from the matroid base polytope of \bar{M}_0 correspond to a nested family of sets. The system $Ax = b$ must contain some constraint corresponding to a matroid $M' \in \mathcal{M}$. Otherwise, we would have a full-rank system consisting of constraints coming from only one matroid, namely \bar{M}_0 , which would yield a unique integral solution; but x^* is not integral. Furthermore, for a matroid $M' = (N', \mathcal{I}') \in \mathcal{M}$, the only constraint of $P_{\mathcal{I}'}$ that can be x^* -tight corresponds to N' , as otherwise, M' would have been refined in step 7. So a matroid $M' \in \mathcal{M}$ gives rise to at most one row of A , which we will also refer to as the row *corresponding* to M' . Let $\emptyset \subsetneq S_1 \subsetneq \dots \subsetneq S_p \subseteq \bar{N}_0$ denote the nested family of sets that give rise to the constraints of \bar{M}_0 in our full-rank system.

Consider the following token-counting argument. Each $e \in \bar{N}_0$ gives $x^*(e)$ tokens to the row of A corresponding to the smallest set S_ℓ containing e (if one exists). It also supplies $(1 - x^*(e))/q_{M'}$ tokens to every row corresponding to a matroid $M' \in \mathcal{M}$ whose ground set contains e . Since the q -values satisfy (2), for each $e \in \bar{N}_0$ the sum of the amount of tokens supplied by e to the rows of A is at most 1. We claim that there is some $e \in \bar{N}_0$ for which this sum is strictly less than 1.

Before proving the claim, we show that it indeed implies the desired result. It follows from the claim that the total amount of tokens distributed is strictly smaller than the number of columns of A . Since $Ax = b$ is a full-rank system and hence A is a square matrix, this implies that there must exist a row of A that receives strictly less than 1 token unit.

Note that every row of A corresponding to a set S_ℓ receives $x^*(S_\ell) - x^*(S_{\ell-1})$ tokens, where $S_0 := \emptyset$. This is positive (since $x^*(e) > 0$ for all e , and $S_{\ell-1} \subsetneq S_\ell$) and an integer (since $x^*(S_i) = r_{\bar{M}_0}(S_i)$ for each $i \in [p]$), and thus at least 1. Along with the observation in the preceding paragraph, this implies that there exists a row corresponding to a matroid $M' = (N', \mathcal{I}') \in \mathcal{M}$ that receives less than 1 token unit; thus $|N'| - x^*(N') < q_{M'}$. Since the rank function $r_{M'}$ of M' is integral, and $r_{M'}(N') \geq x^*(N')$, we have $|N'| - r_{M'}(N') \leq q_{M'} - 1$ as desired.

Finally, we prove the claim that some $e \in \bar{N}_0$ supplies strictly less than one token unit. If every element supplies *exactly* one token unit, then it must be that:

- (i) $S_p = \bar{N}_0$,
- (ii) inequality (2) is tight for all $e \in \bar{N}_0$, and

- (iii) for every $e \in \bar{N}_0$, every matroid $M' = (N', \mathcal{I}')$ in \mathcal{M} with $e \in N'$ gives rise to a row.

Let $A_{M'}$ be the row of A corresponding to matroid $M' \in \mathcal{M}$. Then

$$\sum_{M' \in \mathcal{M}} \frac{1}{q_{M'}} \cdot A_{M'} = \chi^{\bar{N}_0},$$

which is the row of A corresponding to the constraint of \bar{M}_0 for the set S_p . This contradicts that A has full rank. \square

Next, we observe that the algorithm does terminate.

Lemma 8. *Algorithm 1 terminates after at most $\sum_{i=1}^k |N_i| \leq k|N|$ iterations of the main loop.*

Proof. Let us view a refinement operation as replacing a matroid $M' \in \mathcal{M}$ with a matroid M'_1 , and adding a new matroid M'_2 to \mathcal{M} . Then each matroid $M_i = (N_i, \mathcal{I}_i)$ in our input creates (directly or indirectly) at most $|N_i| - 1$ additional matroids, as the refinement of a matroid consists of two matroids with disjoint and nonempty ground sets.

Since the number of original and created matroids is at most $\sum_{i=1}^k |N_i|$, and a matroid is dropped in each iteration of the main loop, there can be at most $\sum_{i=1}^k |N_i|$ iterations in total. \square

We now complete the proof of Theorem 2.

Lemma 9. *The set R returned by Algorithm 1 satisfies the properties stated in Theorem 2.*

Proof. We first observe that if we consider the value $w^T x^* + w(R)$ in each iteration after step 3, this can only increase as the algorithm progresses. Indeed, when we update \bar{M}_0 and \mathcal{M} (via deletions, contractions, refinements, or dropping matroids), x^* restricted to the new ground set remains feasible for (LP_{mat}) for the new instance. This is immediate for deletions and contractions, and if a matroid is dropped; it holds for refinements due to Lemma 5. So if the optimal value of (LP_{mat}) decreases between two executions of line 3 of Algorithm 1, then this is only because we contract elements with $x^*(e) = 1$, which we include in R . Thus the weight of the returned set is at least the weight of the initial fractional solution, and property (i) holds.

We prove that the following invariant is maintained throughout the algorithm:

For any basis B of \bar{M}_0 that is $q_{M'}$ -independent for all $M' \in \mathcal{M}$, $R \cup B$ is a basis of M_0 that is q_i -independent in M_i for each $i \in [k]$.

This trivially holds at the start of the algorithm, and once \bar{N}_0 is reduced to the empty set and the algorithm terminates, it immediately implies properties (ii) and (iii).

The effect of deleting an element e in step 4 is simply to restrict to choices of B which do not contain e ; and the effect of contracting an element e in step 5 is to restrict to choices of B which do. When a matroid $M' \in \mathcal{M}$ is replaced by a refinement M'_1 and M'_2 , any set B that is $q_{M'}$ -independent in M'_1 and M'_2 is $q_{M'}$ -independent in M' by Lemma 6. So in all these cases, assuming the invariant holds before the operation, it will still hold afterwards.

All that remains is to consider the situation when a matroid $M' = (N', \mathcal{I}') \in \mathcal{M}$ is dropped in step 11. Then $r_{M'}(N') + q_{M'} - 1 \geq |N'|$, which means that we can partition N' into a basis (which has size $r_{M'}(N')$) and at most $q_{M'} - 1$ singletons. All singletons are independent in M' because $x_e^* > 0$ for all $e \in N'$. Thus N' is $q_{M'}$ -independent, and hence so is any $B \subseteq \bar{N}_0$. This means that if B satisfies the conditions of the invariant before dropping M' , it still does so afterwards, and hence the invariant still holds. \square

3 Further applications and extensions

3.1 Generalized matroidal degree-bounded spanning tree (GMDST)

In this problem, we are given an undirected graph $G = (V, E)$ with edge costs $c \in \mathbb{R}^E$, disjoint node-sets S_1, \dots, S_k , and matroids $M_i = (\delta(S_i), \mathcal{I}_i)$ for all $i \in [k]$, where $\delta(S_i)$ is the set of edges of G that cross S_i . We want to find a spanning tree T of minimum cost such that $T \cap \delta(S_i) \in \mathcal{I}_i$ for all $i \in [k]$. This generalizes the matroidal degree-bounded MST problem considered by [25], wherein each node $\{v\}$ is an S_i set. Clearly, each edge belongs to at most 2 ground sets of the matroids $\{M_i\}_{i \in [k]}$. Thus, by taking M_0 to be the graphic matroid and setting $w = -c$, Theorem 2 leads to a tree T of cost at most the optimum such that $T \cap \delta(S_i)$ is 2-independent in M_i for all $i \in [k]$.

We remark that, whereas [25] obtains an $O(1)$ -additive violation of the matroid constraints for the matroidal degree-bounded MST problem, such a polytime additive guarantee is not possible for GMDST unless $P = NP$; see Section 4 and the discussion after the proof of Theorem 15.

3.2 Extension to knapsack constraints

We can consider a generalization of (1) where, in addition to the matroids M_0, \dots, M_k (over subsets of N) and the weight vector $w \in \mathbb{R}^N$, we have a family $Cx \leq d$ of t knapsack constraints, where $C \in \mathbb{R}_{\geq 0}^{t \times N}$ and $d \in \mathbb{R}_{\geq 0}^t$. The goal is to find a maximum-weight set R such that $R \in \mathcal{B}_0 \cap \mathcal{I}_1 \cap \dots \cap \mathcal{I}_k$ and $C\chi^R \leq d$.

Again, we consider the following natural LP-relaxation for this problem:

$$\max \left\{ w^T x : x \in \mathbb{R}_{\geq 0}^N, \quad x \in P_{\mathcal{B}_0}, \right. \\ \left. x|_{N_i} \in P_{\mathcal{I}_i} \quad \forall i \in [k], \quad Cx \leq d \right\}. \tag{LP}_{\text{matkn}}$$

We show that Theorem 2 extends to yield the following result.

Theorem 10. *Let $q_1, \dots, q_k \in \mathbb{Z}_{\geq 1}$ be such that*

$$\sum_{i \in [k]: e \in N_i} \frac{1}{q_i} + \sum_{i \in [t]} C_{ie} \leq 1 \text{ for all } e \in N. \quad (4)$$

If $(\text{LP}_{\text{matkn}})$ is feasible, then one can efficiently compute $R \subseteq N$ such that

- (i) $R \in \mathcal{B}_0$;
- (ii) $w(R) \geq \text{OPT}_{\text{LP}_{\text{matkn}}}$;
- (iii) R is q_i -independent in M_i for all $i \in [k]$; and
- (iv) $\sum_{e \in R} C_{ie} \leq d_i + 1$ for all $i \in [t]$.

This theorem yields additive violations of the knapsack constraints by only 1, but (4) is correspondingly a rather strong condition. The theorem can be usefully applied to rescalings of the knapsack constraints in order to satisfy (4). We discuss this in detail after proving the theorem.

Proof. The algorithm leading to Theorem 10 is quite similar to Algorithm 1, and so is its analysis, and we therefore highlight the crucial changes without replicating the proof steps that remain identical.

In the algorithm, whenever we contract an element e , we now update $d_i \leftarrow d_i - C_{ie}$ for every index $i \in [t]$ corresponding to a knapsack constraint that has not yet been dropped. After performing all possible deletions, contractions, and refinements, we now either drop a matroid $M' \in \mathcal{M}'$ in step 11 as before, or we drop a knapsack constraint $\sum_{e \in \bar{N}_0} C_{ie} x(e) \leq d_i$ for some $i \in [t]$ such that $\sum_{e \in \bar{N}_0} C_{ie} (1 - x^*(e)) \leq 1$.

To prove that the modified algorithm is valid, we need only argue that we can always drop a matroid constraint, or a knapsack constraint in step 11 (modified as above). This follows from the same token-counting argument as in the proof of Lemma 7. Recall that if $Ax = b$ is a full-rank subsystem of $(\text{LP}_{\text{matkn}})$ consisting of linearly independent x^* -tight constraints, then we may assume that the rows of A corresponding to the \bar{M}_0 -constraints form a nested family $\mathcal{C} \subseteq 2^{\bar{N}_0}$. We define a token-assignment scheme, where each $e \in \bar{N}_0$ supplies $x^*(e)$ tokens to the row of A corresponding to the smallest set in \mathcal{C} containing e (if one exists), and $(1 - x^*(e))/q_{M'}$ to each row arising from a matroid $M' \in \mathcal{M}$ in our collection whose ground set contains e . *Additionally*, every $e \in \bar{N}_0$ now also supplies $C_{ie}(1 - x^*(e))$ tokens to each row of A arising from a knapsack constraint $i \in [t]$. Under this scheme, as before, given the constraint on our q -values, it follows that every $e \in \bar{N}_0$ supplies at most 1 token unit. Also, as before, each row of A corresponding to a \bar{M}_0 constraint receives at least 1 token unit. So either there is some row coming from a matroid in \mathcal{M} that receives strictly less than 1 token unit, or there must be some row of A corresponding to a knapsack constraint that receives at most 1 token unit; the latter case corresponds to a knapsack constraint indexed by $i \in [t]$ with $\sum_{e \in \bar{N}_0} C_{ie}(1 - x^*(e)) \leq 1$. (Note that if all knapsack constraints have already been dropped, then we indeed have a matroid in \mathcal{M} receiving strictly less than one token as already shown in the proof of Lemma 7.)

The proof of parts (i)–(iii) is exactly as before. To prove part (iv), consider the i -th knapsack constraint. Note that the only place where we possibly introduce a violation in the knapsack constraint is when we drop the constraint. If x^* is the optimal solution just before we drop the constraint, then we know that $\sum_{e \in \bar{N}_0} C_{ie} x^*(e) \leq d_i$. (Note that d_i refers to the updated budget.) It follows that if S denotes the set of elements included from this residual ground set \bar{N}_0 , then the additive violation in the knapsack constraint is

$$\left(\sum_{e \in S} C_{ie} \right) - d_i \leq \left(\sum_{e \in \bar{N}_0} C_{ie} \right) - d_i \leq \sum_{e \in \bar{N}_0} C_{ie} (1 - x^*(e)) \leq 1,$$

where the second inequality follows from $\sum_{e \in \bar{N}_0} C_{ie} x^*(e) \leq d_i$. \square

We now show how the freedom to rescale the knapsack constraints can be used to obtain a more flexible version of Theorem 10 where one can trade off the violation of all involved matroid and knapsack constraints.

Corollary 11. *Let C' be obtained from C by scaling each row so that $\max_{e \in N} C'_{ie} = 1$ for all $i \in [t]$. Let $q_1, \dots, q_k, p_1, \dots, p_t \in \mathbb{Z}_{\geq 1}$ be such that*

$$\sum_{i \in [k]: e \in N_i} \frac{1}{q_i} + \sum_{i \in [t]} \frac{C'_{ie}}{p_i} \leq 1 \quad \forall e \in N. \quad (5)$$

Then if $(\text{LP}_{\text{matkn}})$ is feasible, one can efficiently compute $R \subseteq N$ such that

- (i) $R \in \mathcal{B}_0$;
- (ii) $w(R) \geq \text{OPT}_{\text{LP}_{\text{matkn}}}$;
- (iii) R is q_i -independent in M_i for all $i \in [k]$; and
- (iv) $\sum_{e \in R} C_{ie} \leq d_i + p_i \cdot \max_{e \in N} C_{ie}$ for all $i \in [t]$.

A simpler and less precise condition than (5) is

$$\sum_{i \in [k]: e \in N_i} \frac{1}{q_i} + \sum_{i \in [t]: C_{ie} > 0} \frac{1}{p_i} \leq 1 \quad \forall e \in N. \quad (6)$$

This condition clearly implies (5) since $C'_{ie} \leq 1$ for all $i \in [t], e \in N$. To see the difference between (5) and (6), consider the special case where $p_i = p$ for all $i \in [t]$. Then (loosely speaking) condition (6) requires that p be proportional to the maximum number of knapsack constraints an element participates in, a quantity that is sometimes called the ℓ_0 -column-sparsity of C . But Corollary 11 using the full strength of (5) shows that p can be chosen proportional to the maximum ℓ_1 -norm of any column of the normalized matrix C' . In general, this can be much smaller than the ℓ_0 -column-sparsity of C .

Proof of Corollary 11. Consider the system $\tilde{C}x \leq \tilde{d}$ obtained by scaling the i -th knapsack constraint by $\alpha_i := p_i \cdot \max_{e \in N} C_{ie}$ for all $i \in [t]$; that is, we have $\tilde{C}_{ie} = C_{ie}/\alpha_i = C'_{ie}/p_i$ for all $i \in [t], e \in N$, and $\tilde{d}_i = d_i/\alpha_i$ for all $i \in [t]$. Notice

that the conditions of Theorem 10 are fulfilled for the scaled instance, because for any $e \in N$, we have by (5) that

$$\sum_{\substack{i \in [k]: \\ e \in N_i}} \frac{1}{q_i} + \sum_{i \in [k]} \tilde{C}_{ie} = \sum_{\substack{i \in [k]: \\ e \in N_i}} \frac{1}{q_i} + \sum_{i \in [t]} \frac{C'_{ie}}{p_i} \leq 1.$$

Applying Theorem 10 to the instance with the scaled knapsack constraints, we obtain a set R guaranteed by Theorem 10 that clearly satisfies (i)–(iii). Moreover, (iv) holds because the additive violation of 1 for each scaled knapsack constraint translates to an additive violation of $\alpha_i = p_i \cdot \max_{e \in N} C_{ie}$ for the original instance. \square

Applications and refinements. A variety of settings considered in the literature can be viewed as special cases of the above setup. Specifically, Grandoni et al. [13] consider the *t-budgeted matroid independent-set* (or basis) and *t-budgeted matroid intersection* problems, which in turn generalize various problems, such as *t-budgeted spanning trees* and *t-budgeted bipartite matchings* (see [13] for an extensive discussion of work on these and other related problems). In our setup, the above two problems correspond to the cases where $k = 0$ and $k = 1$, respectively, and we have t knapsack constraints. (Recall we have k matroids M_1, \dots, M_k in addition to M_0 ; moreover, as discussed in the proof of Corollary 3, while we state our problem in terms of finding a basis of M_0 , this can be used to model the setting where we seek an independent set of M_0 .)

The *t-budgeted matroid basis* problem also captures the problem of minimizing makespan on unrelated machines: matroid M_0 encodes that every job is assigned to a machine, and the knapsack constraints encode that the load on each machine is at most a given makespan bound. We use this setup to illustrate the utility of having bounds depending on the ℓ_1 -column-sparsity of the knapsack constraints. Recently, Chakrabarty and Swamy [4] considered a more general load-balancing problem where one seeks to minimize the norm of the machine-loads vector under a given monotone, symmetric norm. They show that this problem reduces to the problem of finding a min-cost assignment of jobs to machines subject to *multiple* load constraints for each machine. In this reduction, the load constraints for a fixed machine involve a nested family of job-sets and have geometrically increasing budgets. Therefore, after normalizing the load constraints by the budgets, the ℓ_1 norm of each column becomes a constant. Hence, our results yield a constant-factor violation of the machine-load constraints, and [4] show that this leads to an $O(1)$ -approximation for the minimum-norm load balancing problem.

The work of [6, 13] yields (deterministic or randomized) polynomial time approximation schemes for *t-budgeted matroid independent-set* and *t-budgeted matroid intersection* problems, when t is a constant. While a direct application of Corollary 11 results in a violation of both the matroid independence and knapsack constraints, we show that this can be improved. First, for $k \leq 2$, we can translate approximate matroid independence into an approximation in the

objective, as shown in the proof of Theorem 1. Second, we show below that by using a standard enumeration idea and insights from [13], we can eliminate the violation in the knapsack constraints as well, when $k, t = O(1)$. Consequently, we obtain the following guarantees, when $t = O(1)$.

- A PTAS for t -budgeted matroid independent set; this matches the guarantee in [13].
- A $(3 + \delta)$ -approximation for t -budgeted 3-matroid intersection (i.e., $k = 2$), wherein we seek a maximum-weight common independent set of 3 matroids that satisfies t knapsack constraints (Theorem 14). This improves upon the constant-factor approximation obtained by [7] for this problem. (Our result also implies a $(2 + \delta)$ -approximation for t -budgeted 2-matroid intersection, but here a PTAS is known, as noted earlier.)

In the sequel, we prove the second result. We do not explicitly prove the first result as a similar result is already known, but this guarantee follows from Theorem 13 below. We focus on the version where we seek a set that is independent in all matroids; for the version where we seek a basis of M_0 , no true approximation is possible even when M_0 is the only matroid and $t = 2$ [13]. Recall that we are given matroids $M_i = (N_i, \mathcal{I}_i)$ for $i = 0, \dots, k$, where $N_i \subseteq N$ for all $i = 0, \dots, k$, a weight vector $w \in \mathbb{R}_{\geq 0}^N$, and t knapsack constraints $Cx \leq d$. We want to maximize $w(R)$ subject to $R \in \mathcal{I}_0 \cap \mathcal{I}_1 \cap \dots \cap \mathcal{I}_k$ and $C\chi^R \leq d$. Let Z^* denote an optimal solution to the problem, and $OPT = w(Z^*)$ denote the optimal value. The following simple lemma will be useful.

Lemma 12 (Paraphrased from [13]). *Let $\ell \in \mathbb{R}_{\geq 0}^N$, $L \geq 0$, and $\delta \in (0, 1]$. Further, let $S \subseteq N$ be such that $\ell(S) \leq L$. Then we can efficiently find $S' \subseteq S$ such that $w(S') \leq \delta w(S) + \max_{e \in S} w_e$ and $\ell(S \setminus S') \leq (1 - \delta)L$.*

Proof. We assume without loss of generality that $\ell_e > 0$ for every $e \in S$; otherwise, we could simply use the set S' obtained by applying the lemma to the set $\{e \in S : \ell_e > 0\}$ instead of S .

We sort the elements of S in increasing order of w_e/ℓ_e . Considering elements of S in this sorted order, let S' be the smallest prefix (which might be \emptyset) such that $\ell(S \setminus S') \leq (1 - \delta)L$. If $S' = \emptyset$, then we are done, so assume otherwise. Let e' be the last element in S' under the sorted order, so $\min_{e \in S \setminus S'} w_e/\ell_e \geq w_{e'}/\ell_{e'} = \max_{e \in S'} w_e/\ell_e$. Let $S'' = S' \setminus \{e'\}$.

Due to the ordering of elements, we have $w(S \setminus S'')/\ell(S \setminus S'') \geq w(S)/\ell(S) \geq w(S)/L$. Due to the choice of S' , we have $\ell(S \setminus S'') > (1 - \delta)L$. Therefore, $w(S \setminus S'') \geq w(S) \cdot \ell(S \setminus S'')/L > (1 - \delta)w(S)$. It follows that $w(S'') < \delta w(S)$, and hence $w(S') < \delta w(S) + w_{e'} \leq \delta w(S) + \max_{e \in S} w_e$. \square

We now show how to avoid violation of the knapsack constraints.

Theorem 13. *Let $q_1, \dots, q_k \in \mathbb{Z}_{\geq 1}$, and $\epsilon > 0$ satisfy*

$$\sum_{i \in [k]: e \in N_i} \frac{1}{q_i} + \epsilon |\{i \in [t] : C_{ie} > 0\}| \leq 1 \text{ for all } e \in N.$$

In time $\text{poly}(|N|^{O(t^2/\epsilon^2)})$, one can compute $R \subseteq N$ such that

- (i) $R \in \mathcal{I}_0$;
- (ii) $w(R) \geq (1 - 2\epsilon)OPT$;
- (iii) R is q_i -independent in M_i for all $i \in [k]$; and
- (iv) $\sum_{e \in R} C_{ie} \leq d_i$ for all $i \in [t]$.

Proof. We utilize a standard enumeration idea to reduce the violation of the knapsack constraints to a multiplicative $(1 + O(\epsilon/t))$ -factor, and then use insights from [13] to eliminate this violation altogether. The idea is to first guess all elements included in the optimal solution Z^* that have large w_e -weight. We modify the matroids and the knapsack constraints to account for including these elements. We now scale down the residual budgets of the knapsack constraints. Since the maximum weight of any element in the residual instance is small, Lemma 12 allows one to argue that there is some $Z' \subseteq Z^*$ that satisfies these scaled budgets and still has large value. Now we guess the elements of Z' that have large cost compared to the residual budget for any of the knapsack constraints, and again modify the matroids and knapsack constraints accordingly. Applying Corollary 11 to this instance, since, for every knapsack constraint, the maximum cost of any element is small compared to its residual budget, we obtain a solution that violates the scaled knapsack constraints by a small factor. Due to our scaling of budgets, this translates to no violation of the original knapsack constraints. We now furnish the details.

We assume that $|Z^*| > \lceil t/\epsilon \rceil$; otherwise, we can find an optimal solution by brute force in $\text{poly}(|N|^{O(t/\epsilon)})$ time. For $S \subseteq N$ and $i \in [t]$, we use $C_i(S)$ to denote $\sum_{e \in S} C_{ie}$. We guess the set A of $\lceil t/\epsilon \rceil$ elements included in Z^* that have the largest w_e -weight. We prove below that there exists $Z' \subseteq Z^* \setminus A$ such that $w(Z') \geq (1 - 2\epsilon)OPT - w(A)$, and $C_i(Z') \leq (1 - \epsilon/t)(d_i - C_i(A))$ for every $i \in [t]$. For every $i \in [t]$, we guess the set B_i of at most t/ϵ^2 elements from Z' for which $C_{ie} > \frac{\epsilon^2}{t} \cdot (d_i - C_i(A))$. (More precisely, by guessing, we mean that we enumerate all possible choices for A and the B_i s. For each such choice, we execute the steps described below. The analysis shows that for the correct choice of A and B_i s, the set R returned has the desired properties.)

Let $F := B_1 \cup \dots \cup B_t$. We modify the matroids by contracting all the elements in $A \cup F$, and deleting every element $e \in N \setminus (A \cup F)$ for which $w_e > \min_{e' \in A} w_{e'}$ or there exists $i \in [t]$ with $C_{ie} > \frac{\epsilon^2}{t} \cdot (d_i - C_i(A))$. Let N' denote the remaining set of elements. So by construction, we have: (a) $|A \cup F| \leq \lceil t/\epsilon \rceil + t^2/\epsilon^2$; (b) $N' \cap (A \cup F) = \emptyset$; and (c) $C_{ie} \leq \frac{\epsilon^2}{t} \cdot (d_i - C_i(A))$ for all $i \in [t], e \in N'$. For each $i \in [t]$, we modify the i -th knapsack constraint so that its budget is $d'_i = (1 - \epsilon/t)(d_i - C_i(A)) - C_i(F)$ and we only consider elements from N' . Call this the residual instance.

We apply Corollary 11 to this residual instance with the q_i s given in the theorem statement, and taking $p_i = 1/\epsilon$ for all $i \in [t]$. Notice that these values satisfy (5). Note that mimicking the construction as used in the proof of Corollary 3 shows that Corollary 11 also holds when we require only an independent

set of M_0 ; let R' be the set returned by this variant of Corollary 11. We return $R = R' \cup A \cup F$.

Property (a) above implies that there is $O(|N|^{O(t^2/\epsilon^2)})$ choices for A and the B_i s, which gives the stated running time.

We now prove that R satisfies all the desired properties. By the construction of the residual instance, it follows that $R \in \mathcal{I}_0$, and R is q_i -independent in M_i for all $i \in [k]$.

Let OPT' denote the optimal value for the residual instance. We prove that $OPT' \geq (1 - 2\epsilon)OPT - w(A) - w(F)$, which implies that $w(R') \geq OPT'$ (due to guarantee (ii) of Corollary 11), and hence, $w(R) = w(R') + w(A) + w(F) \geq (1 - 2\epsilon)OPT$.

Consider the set $Z = Z^* \setminus A$. Consider the i -th knapsack constraint. We have $C_i(Z) \leq d_i - C_i(A)$. Therefore, taking $\delta = \epsilon/t$, ℓ to be the i -th row of C , and $L = d_i - C_i(A)$ in Lemma 12, we can find $Z_i \subseteq Z$ such that $w(Z_i) \leq \epsilon w(Z)/t + \max_{e \in Z} w_e$ and $C_i(Z \setminus Z_i) \leq (1 - \epsilon/t)(d_i - C_i(A))$. Observe that $\max_{e \in Z} w_e \leq \min_{e \in A} w_e \leq \epsilon OPT/t$, so $w(Z_i) \leq 2\epsilon OPT/t$. So the set $Z' = Z \setminus (Z_1 \cup \dots \cup Z_t)$ satisfies $w(Z') \geq (1 - 2\epsilon)OPT - w(A)$ and $C_i(Z') \leq (1 - \epsilon/t)(d_i - C_i(A))$ for all $i \in [t]$. Therefore, $Z' \setminus F$ is feasible for the residual instance, and $OPT' \geq w(Z') - w(F) \geq (1 - 2\epsilon)OPT - w(A) - w(F)$.

Finally, for every $i \in [t]$ we have $C_i(R') \leq d'_i + \frac{1}{\epsilon} \cdot \max_{e \in N'} C_{ie}$ from guarantee (iv) of Corollary 11. By property (c) above, we then have $C_i(R') \leq d'_i + \frac{\epsilon}{t} \cdot (d_i - C_i(A)) = d_i - C_i(A) - C_i(F)$, and hence, $C_i(R) \leq d_i$. \square

Theorem 14. *Let $0 < \delta \leq 1/6$. There is a $(3 + O(\delta))$ -approximation algorithm for t -budgeted 3-matroid intersection with running time $\text{poly}(|N|^{O(t^4/\delta^2)})$, where N is the common ground set of the matroids and the knapsack constraints.*

Proof. Let $M_i = (N, \mathcal{I}_i)$ for $i = 0, 1, 2$ be the given matroids, $Cx \leq d$ be the t knapsack constraints, and $w \in \mathbb{R}_{\geq 0}^N$ be the given weight vector. Let OPT denote the optimal value. We apply Corollary 13 to this instance taking $q_1 = q_2 = 3$, and $\epsilon = \delta/t$. Since $\delta \leq 1/3$, clearly these values satisfy the condition in Corollary 13. Therefore, in time $\text{poly}(|N|^{O(t^2/\epsilon^2)}) = \text{poly}(|N|^{O(t^4/\delta^2)})$, we obtain a set $R \in \mathcal{I}_0$ that is 3-independent in M_1, M_2 , satisfies all the knapsack constraints, and has weight $w(R) \geq (1 - 2\epsilon)OPT$. As in the proof of Theorem 1, we can now extract a subset $R' \subseteq R$ that is independent in all the matroids, and such that $w(R') \geq (1 - 2\epsilon)OPT/3 \geq OPT/(3 + 9\epsilon)$. The last inequality follows since $1 - 2\epsilon \geq (1 + 3\epsilon)^{-1}$ as $\epsilon \leq \delta \leq 1/6$. \square

4 Impossibility of achieving small additive violations

We show that Theorem 2 for problem (1) cannot be strengthened to yield a basis of M_0 that has small additive violation for the matroid constraints of M_1, \dots, M_k , even when $k = 2$.

We first define additive violation precisely. Given a matroid $M = (N, \mathcal{I})$, we say that a set $R \subseteq N$ is μ -additively independent in M if $|R| - r_M(R) \leq \mu$; equivalently, we can turn R into an independent set in M by removing at most μ elements. Unlike results for degree-bounded spanning trees, or matroidal degree-bounded MST [25], we show that small additive violation is not possible in polytime (assuming $P \neq NP$) even for the special case of (1) where $k = 2$, so we seek a basis of M_0 that is independent in M_1, M_2 .

Theorem 15. *Let $f(n) = O(n^{1-\varepsilon})$, where $\varepsilon > 0$ is a constant. Suppose we have a polytime algorithm \mathcal{A} for (1) that returns a basis B of M_0 satisfying $|B| \leq r_{M_i}(B) + f(|N|)$ for $i = 1, 2$. Then we can find in polytime a basis of M_0 that is independent in M_1, M_2 .*

The problem of finding a basis of M_0 that is independent in M_1, M_2 is NP -hard, as shown by an easy reduction from the directed Hamiltonian path problem. Thus, Theorem 15 shows that it is NP -hard to obtain an additive violation for problem (1) that is substantially better than linear violation.

Proof of Theorem 15. Choose t large enough so that $t > 2f(t|N|)$. Since $f(n) = O(n^{1-\varepsilon})$, this is achieved by some $t = \text{poly}(|N|)$. For each $i \in \{0, 1, 2\}$, let M'_i be the direct sum of t copies of M_i . Let N' be the ground set of these matroids, which consists of t disjoint copies of N , which we label N_1, \dots, N_t .

Clearly, the instance (M'_0, M'_1, M'_2) is feasible if and only if the original instance is feasible. Suppose that running \mathcal{A} on the replicated instance yields a basis R' of M'_0 that has the stated additive violation for the matroids M'_1, M'_2 . Hence, there are two sets $Q_1, Q_2 \subseteq R'$ with $|Q_1|, |Q_2| \leq f(t|N|)$, such that $R' \setminus Q_i$ is independent in M'_i for $i = 1, 2$. Hence, $R' \setminus (Q_1 \cup Q_2)$ is independent in both M'_1 and M'_2 . Because $|Q_1 \cup Q_2| \leq 2f(t|N|) < t$, we have by the pigeonhole principle that there is one $j \in [t]$ such that $(Q_1 \cup Q_2) \cap N_j = \emptyset$. This implies that $R = R' \cap N_j = (R' \setminus (Q_1 \cup Q_2)) \cap N_j$, when interpreted on the ground set N , is independent in both M_1 and M_2 . Moreover, the elements of R , when interpreted on the ground set N , are a basis in M_0 because R' is a basis in M'_0 . Hence, R is the desired basis without any violations. \square

Theorem 15 holds whenever M_0, M_1, M_2 come from classes of matroids that are closed under direct sums. Its proof easily extends to the setting where we have $k + 1$ matroids M_0, \dots, M_k , each of which comes from a class of matroids that is closed under taking direct sums, and shows that achieving a much-better-than-linear additive violation for problem (1) would enable one to obtain a basis of M_0 that is independent in M_1, \dots, M_k . Consequently, if the latter problem is NP -hard for the given classes of matroids, then obtaining an additive violation that is much better than linear violation is NP -hard. In particular, this implies that for the generalized matroidal degree-bounded spanning tree (GMDST) problem considered in Section 3, since the family of graphic matroids is closed under taking direct sums, and (GMDST) generalizes the NP -hard degree-bounded spanning tree problem, it is NP -hard to achieve an additive violation substantially better than linear for the matroid constraints.

5 Conclusions

We presented a new iterative rounding procedure applicable to the intersection of several matroid polytopes and knapsack constraints. A key technical component of our procedure is the refinement of matroids. In contrast with typical iterative rounding procedures, a refinement replaces one matroid constraint by two that, combined, are stronger than the original one. As such, this does not correspond to a relaxation. The purpose of these refinement steps is to make it easier to drop one of the newly-created matroid constraints at a later stage. One key implication of our approach is an LP-relative 2-approximation for 3-matroid intersection. Apart from being the first procedure that achieves a factor-2 approximation, it also settles the integrality gap of the natural linear program for the intersection of three matroids. We also present an extension that enables one to handle both matroid independence and knapsack constraints. Moreover, we show that our rounding framework allows one to capture various other natural problem settings in a unified way.

A natural open question is to obtain an LP-relative $(k - 1)$ -approximation for the intersection of k matroids. It does not seem straightforward to adapt our approach to this setting, because our method for 3-matroid intersection first obtains a set that is independent in one of the matroids and is 2-independent in the other two; it then crucially exploits the fact that the intersection of the matroid polytopes of the other two matroids is integral. This enables us to scale down our solution by a factor of 2 and observe that it must be a convex combination of sets that are independent in all three matroids. Still, perhaps a variation of our overall scheme, namely first obtaining an infeasible set and later correcting it, could be a promising direction.

Acknowledgements. We are thankful to Lap Chi Lau for pointing us to relevant literature, and to the anonymous referees for helpful suggestions for improving the exposition.

References

1. R. Aharoni and E. Berger. The Intersection of a Matroid and a Simplicial Complex. *Transactions of the American Mathematical Society*, 358(11): 4895–4917 (2006).
2. N. Bansal, R. Khandekar, J. Könemann, V. Nagarajan, and B. Peis. On generalizations of network design problems with degree bounds. *Mathematical Programming*, 141(1-2): 479-506 (2013).
3. N. Bansal, R. Khandekar, and V. Nagarajan. Additive guarantees for degree-bounded directed network design. *SIAM Journal on Computing*, 39(4):1413–1431, 2009.
4. D. Chakrabarty and C. Swamy. Approximation algorithms for minimum norm and ordered optimization problems. In *Proceedings of the 51st Annual ACM Symposium on Theory of Computing (STOC)*, pages 126–137, 2019.
5. Y. H. Chan and L. C. Lau. On linear and semidefinite programming relaxations for hypergraphic matching. *Mathematical Programming*, 135:123–148, 2012.

6. C. Chekuri, J. Vondrák, and R. Zenklusen. Multi-budgeted matchings and matroid intersection via dependent rounding. In *Proceedings of the 22nd Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 1080–1097, 2011.
7. C. Chekuri, J. Vondrák, and R. Zenklusen. Submodular function maximization via the multilinear relaxation and contention resolution schemes. *SIAM Journal on Computing*, 43(6):1831–1879, 2014.
8. W. J. Cook, W. H. Cunningham, W. R. Pulleyblank, and A. Schrijver. *Combinatorial Optimization*, 1997.
9. M. Cygan. Improved approximation for 3-dimensional matching via bounded pathwidth local search. In *Proceedings of 54th IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 509–518, 2013.
10. M. Cygan, F. Grandoni, and M. Mastrolilli. How to sell hyperedges: The hypermatching assignment problem. In *Proceedings of the 24th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 342–351, 2013.
11. M. L. Fisher, G.L. Nemhauser, and L. A. Wolsey. An analysis of approximations for maximizing submodular set functions - II. *Mathematical Programming Study*, 8:73–87, 1978.
12. Z. Füredi. Maximum degree and fractional matchings in uniform hypergraphs. *Combinatorica*, 1(2):155–162, 1981.
13. F. Grandoni, R. Ravi, M. Singh, and R. Zenklusen. New approaches to multi-objective optimization. *Mathematical Programming*, 146(1-2):525–554, 2014.
14. A. Gupta, A. Roth, G. Schoenebeck, and K. Talwar. Constrained non-monotone submodular maximization: offline and secretary algorithms. In *Proceedings of the 6th International Conference on Internet and Network Economics (WINE)*, pages 246–257, 2010.
15. M. M. Halldórsson. Approximating discrete collections via local improvements. In *Proceedings of the 6th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 160–169, 1995.
16. C. A. J. Hurkens and A. Schrijver. On the size of systems of sets every t of which have an SDR, with an application to the worst-case ratio of heuristics for packing problems. *SIAM Journal on Discrete Mathematics*, 2(1):68–72, 1989.
17. T. Király, L. C. Lau, and M. Singh. Degree bounded matroids and submodular flows. *Combinatorica*, 32(6): 703-720 (2012).
18. L. C. Lau, R. Ravi, and M. Singh. *Iterative Methods in Combinatorial Optimization*. Cambridge University Press, New York, NY, USA, 1st edition, 2011.
19. J. Lee, V. Mirrokni, V. Nagarajan, and M. Sviridenko. Maximizing nonmonotone submodular functions under matroid or knapsack constraints. *SIAM Journal on Discrete Mathematics*, 23(4):2053–2078, 2010.
20. J. Lee, M. Sviridenko, and J. Vondrák. Submodular maximization over multiple matroids via generalized exchange properties. *Mathematics of Operations Research*, 35(4):795–806, 2010.
21. A. Linhares, N. Olver, C. Swamy, and R. Zenklusen. Approximate multi-matroid intersection via iterative refinement. In *Proceedings of Integer Programming and Combinatorial Optimization (IPCO)*, pages 299–312, 2019.
22. O. Parekh and D. Pritchard. Generalized hypergraph matching via iterated packing and local ratio. In *Proceedings of Workshop on Approximation and Online Algorithms (WAOA 2014)*, pages 207–223. Springer International Publishing, 2015.
23. A. Schrijver. *Combinatorial Optimization, Polyhedra and Efficiency*. Springer, 2003.
24. M. Singh and L. C. Lau. Approximating minimum bounded degree spanning trees to within one of optimal. *J. ACM*, 62(1): 1:1-1:19 (2015).

25. R. Zenklusen. Matroidal degree-bounded minimum spanning trees. In *Proceedings of the 23rd Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 1512–1521, 2012.